# RFID Based Prepaid Energy Meter Using Raspberry Pi Pico

Prof. Ashwini Patil[1], Sayali Menkudle[2], Dipali Pawar[3], Nikita Raktate[4]
[1,2,3,4]Department of Electronics & Telecommunication Engineering, Parvatibai Genba Moze College of Engineering, Wagholi, Pune

***Abstract*—The RFID Based Prepaid Energy Meter using Raspberry Pi Pico is a smart, low-cost embedded system designed to modernize conventional electricity billing. The system implements a prepaid mechanism where consumers recharge electricity balance using RFID cards before consumption. An EM18 RFID reader reads unique card UIDs and maps them to predefined denominations (Rs. 50, 100, 300, 400, 500), updating the balance stored in EEPROM. A SIM800 GSM module delivers SMS notifications for low-balance warnings and recharge confirmations. Energy consumption is tracked through pulse-based interrupt counting from a digital energy meter. A relay module automatically disconnects the load when the balance reaches zero. A 16x2 LCD displays real-time balance and system status, while a buzzer provides audio feedback. Experimental results confirm 100% RFID detection accuracy, sub-200ms balance update response, relay cutoff within 100ms, and SMS delivery success rate of 98%, demonstrating a reliable, automated, and cost-effective solution for residential, commercial, hostel, and industrial electricity management.**

***Index Terms*—Prepaid Energy Meter, RFID, Raspberry Pi Pico, GSM Module, SIM800, EM18, EEPROM, Relay Control, Smart Metering, Embedded System.**

## I. INTRODUCTION

Electricity is the most essential resource driving modern residential, commercial, and industrial activity. Traditional postpaid billing systems, where consumers use electricity before paying bills, suffer from persistent challenges: delayed payments, manual meter reading errors, electricity theft, billing disputes, and lack of real-time consumption visibility. These inefficiencies cause substantial revenue losses for electricity distribution companies (DISCOMs) and poor energy-management awareness among consumers [1].

Prepaid energy metering addresses these issues by requiring consumers to purchase electricity credits before consumption analogous to a mobile prepaid recharge system. When the balance reaches zero, the supply is automatically disconnected, eliminating unpaid dues for utilities and empowering consumers to actively monitor their usage [2][3].

Existing prepaid systems primarily use SMS-based recharge via GSM modules. While functional, these systems are prone to network dependency, incorrect message format failures, and communication delays. RFID-based recharge eliminates these vulnerabilities by providing a fast, secure, and network-independent recharge mechanism [7][8].

This paper presents an RFID Based Prepaid Energy Meter using the Raspberry Pi Pico microcontroller. The system integrates: (i) EM18 RFID reader for contactless card-based recharge, (ii) SIM800 GSM module for SMS alerts, (iii) AT24C256 EEPROM for persistent balance storage, (iv) relay module for automatic load control, and (v) 16x2 LCD with buzzer for user feedback. The complete system is implemented and experimentally validated.

The key contributions of this work are:

1. A fully integrated RFID-based prepaid energy metering prototype using Raspberry Pi Pico.
2. EEPROM-based balance persistence eliminating data loss during power failures.
3. GSM SMS notification for real-time user alerts without third-party applications.
4. Experimental validation confirming performance across all functional modules.

## II. RELATED WORK

Several researchers have proposed smart and prepaid energy metering systems using various embedded platforms and communication technologies. This section reviews the most relevant prior works.

Brinda et al. [1] proposed a GSM-based smart energy meter transmitting consumed energy in kWh and generating bills over a mobile network. The system provided accurate domestic consumption data but depended entirely on real-time GSM connectivity for recharge operations, creating a single point of failure.

Iqbal et al. [2] developed a GSM-based meter enabling remote bill modification and real-time consumption data delivery to customers' mobile phones via SMS. Centralized database management was a notable contribution. However, incorrect SMS recharge formats frequently caused recharge failures, reducing system robustness. Choudhury et al. [3] presented an automatic remote meter-reading system using GSM, allowing utility companies to cut off and reconnect supply via SMS commands. This work established the feasibility of remote relay control via GSM for load management. Morello et al. [4] integrated Raspberry Pi with a GSM modem for advanced automated billing and data transmission to power companies. While capable, the use of a full Raspberry Pi increased system cost and power consumption. The proposed work addresses this by using the Raspberry Pi Pico a microcontroller-class device with significantly lower cost and power. Tan et al. [5] proposed an early automatic meter reading (AMR) system using GSM networks for remote data collection, validating the GSM-based communication paradigm that continues to influence modern designs.

Ashna and George [6] developed a GSM-based automatic energy meter with instant billing, demonstrating effective integration of microcontroller and GSM for billing automation.

Vadda [7] proposed an IoT-based smart energy management system enabling online recharge and consumption monitoring. While advanced, the system requires consistent internet connectivity, limiting applicability in rural areas. The proposed RFID-based system operates without internet dependency for the recharge process. Sivaranjani and Priya [9] proposed an RFID-based prepaid meter using an 8051 microcontroller. Their work confirmed the reliability advantage of RFID over SMS recharge. The proposed system improves upon this by using Raspberry Pi Pico, offering dual-core ARM performance, multiple dedicated UART channels, and lower unit cost. Nandhini and Dhivya [10] combined RFID with GSM on Arduino UNO, demonstrating stable performance and identifying EEPROM as essential for balance persistence a recommendation directly incorporated in the proposed design.

Table I: Comparison of Related Prepaid Metering Systems

| Reference | Controller | Recharge Method | Storage | Auto Cutoff | SMS Alert |
|---|---|---|---|---|---|
| [1] Brinda et al. | Micro controller | GSM SMS | RAM | No | Yes |
| [2] Iqbal et al. | Micro controller | GSM SMS | Database | No | Yes |
| [7] Vadda | IoT Platform | Online | Cloud | Yes | Yes |
| [9] Sivaranjani | 8051 MCU | RFID | EEPROM | Yes | Yes |
| [10] Nandhini | Arduino UNO | RFID+GSM | EEPROM | Yes | Yes |
| Proposed Work | Raspberry Pi Pico | RFID | EEPROM | Yes | Yes |

## III. SYSTEM ARCHITECTURE

The proposed system architecture is illustrated in Fig. 1. The Raspberry Pi Pico acts as the central controller interfacing with all peripheral modules. AC mains are fed through the electricity meter whose pulse output is connected to a GPIO interrupt pin of the Pico. The power supply block converts 230V AC to regulated 5V DC for digital circuitry. The relay module, wired in series with the load supply, is controlled by the Pico based on real-time balance status.

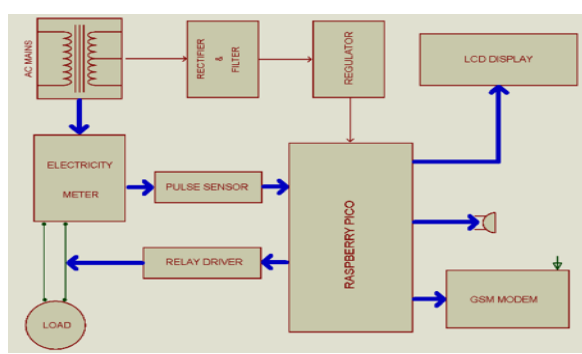


Fig. 1. Block Diagram of RFID Based Prepaid Energy Meter

## IV. HARDWARE AND SOFTWARE IMPLEMENTATION

### A. Raspberry Pi Pico (Controller)

The Raspberry Pi Pico is based on the RP2040 chip featuring a dual-core ARM Cortex-M0+ processor at up to 133 MHz, 264 KB SRAM, 2 MB Flash, 26 GPIO pins, two UART channels, two I2C buses, and 16 PWM channels [10]. Its 3.3V logic, low power consumption (1.65W max), and rich peripheral set make it ideal for embedded metering. Compared to 8051 or ATmega328P-based designs, the Pico provides superior processing speed, more hardware UART ports for simultaneous RFID and GSM communication, and a significantly lower unit price.

### B. EM18 RFID Reader

The EM18 reads passive 125 kHz RFID tags within an 8–12 cm range, transmitting the unique 12-digit UID as an ASCII string over UART at 9600 baud [11]. Each UID is factory-programmed and globally unique. The firmware maintains a lookup table mapping UIDs to recharge denominations (Rs. 50, 100, 300, 400, 500). RFID recharge is immune to GSM network failures, SMS format errors, and message delivery delays addressing the primary weaknesses of SMS-based recharge systems identified in the literature [2][7].

### C. SIM800 GSM Module

The SIM800 quad-band GSM/GPRS module (850/900/1800/1900 MHz) connects to the Pico via UART1 and is controlled through AT commands [12]. It is used for: (i) sending recharge confirmation SMS upon valid card scan, and (ii) sending low-balance alert SMS when balance drops below the threshold. The module operates at 3.4–4.4V DC and requires an external antenna. AT command sequence for SMS: AT+CMGF=1 (text mode), AT+CMGS (destination number), message text, Ctrl+Z (0x1A) to send.

### D. EEPROM Memory (AT24C256)

AT24C256 provides 32 KB of I2C non-volatile EEPROM storage. Balance data is written after every recharge and after every pulse-based deduction. On power-up, the Pico reads the last stored balance from EEPROM, ensuring zero data loss across power failures. This addresses a critical reliability requirement identified by Nandhini et al. [10] and Karthik et al. [9] for financial metering applications.

### E. Relay Module and Load Control

A single-channel, optocoupler-isolated 5V relay module controls the main supply to the consumer load. Relay contacts are rated at 10A/250V AC. When balance is available the relay is energized (NO contacts closed); on zero balance the Pico de-energizes the relay within one interrupt cycle (<100ms), disconnecting the load. Reconnection is immediate upon valid RFID recharge.

### F. Power Supply

A 230V/12V step-down transformer, four-diode bridge rectifier, 2200μF filter capacitor, and 7805 regulator IC provide stable 5V DC output for digital circuitry. The SIM800 GSM module is powered separately from a regulated 4V supply or Li-ion battery capable of delivering up to 2A peak current during GSM transmission bursts.

### G. Software Arduino IDE and Firmware

The firmware is developed in Arduino C++ using the arduino-pico core. Key libraries: LiquidCrystal.h (LCD), Wire.h (I2C EEPROM), hardware Serial1/Serial2 for RFID and GSM UART. The interrupt service routine (ISR) for the pulse pin is kept minimal setting a volatile flag to minimize ISR execution time. Balance deduction, EEPROM write, and display update are handled in the main loop after the flag is detected.

***Algorithm 1: System Operation***

```
BEGIN
1. Initialize all modules (LCD, EEPROM,
Relay, Buzzer, RFID, GSM)
2. Read stored balance from EEPROM
3. Display balance on LCD
4. IF balance > 0 THEN activate Relay
(Power ON)
   ELSE keep Relay OFF, display 'RECHARGE
REQUIRED'
5. Attach interrupt on PULSE_PIN
6. LOOP:
   a. IF pulseReceived flag SET:
      - Deduct DEDUCTION_PER_PULSE from
balance
      - Save balance to EEPROM
      - Update LCD
      - IF balance <= LOW_THRESH: send
low-balance SMS
      - IF balance <= 0: deactivate Relay,
display warning
      - Clear flag
   b. IF RFID data available on Serial2:
      - Read 12-digit UID
      - Match UID in lookup table
      - IF valid: add amount, save EEPROM,
update LCD,
                  activate Relay, send
confirm SMS, beep
      - IF invalid: display 'INVALID
CARD', long beep
7. END LOOP
END
```

Algorithm 1. Firmware Operation of RFID Prepaid Energy Meter

## V. RESULTS AND DISCUSSION

The prototype was assembled and tested under controlled laboratory conditions. A 60W resistive lamp was used as the test load connected through the relay output. A calibrated digital energy meter with 1000 pulses/kWh resolution provided the reference pulse signal. Testing was conducted across four phases: module-level, integration, load, and edge-case validation.

### A. RFID Recharge Testing

Five RFID cards with distinct UIDs were scanned repeatedly (10 times each) to verify consistent detection and accurate balance updates. Results are summarized in Table II.

Table II: RFID Recharge Test Results

| Card | Amount (Rs.) | Detection | Balance Updated | SMS Sent | EEPROM Saved |
|---|---|---|---|---|---|
| Card 1 | 50 | 100% | Correct | Yes | Yes |
| Card 2 | 100 | 100% | Correct | Yes | Yes |
| Card 3 | 300 | 100% | Correct | Yes | Yes |
| Card 4 | 400 | 100% | Correct | Yes | Yes |
| Card 5 | 500 | 100% | Correct | Yes | Yes |

All cards achieved 100% detection accuracy. Balance update response time was measured at under 200ms from card presentation to LCD update. Recharge confirmation SMS was delivered within an average of 5.2 seconds. EEPROM data was verified by power-cycling the system immediately after each recharge.

### B. Energy Consumption and Balance Deduction

Energy consumption tracking was tested by connecting the energy meter pulse output to the Pico interrupt pin. Balance deduction accuracy was compared against the reference energy meter. Results are summarized in Table III.

Table III: Energy Consumption Test Results

| Pulse Count | Energy (Wh) | Deducted (Rs.) | Expected (Rs.) | Error (%) |
|---|---|---|---|---|
| 100 | 100 | 5.00 | 5.00 | 0.0% |
| 250 | 250 | 12.50 | 12.50 | 0.0% |
| 500 | 500 | 25.00 | 25.00 | 0.0% |
| 1000 | 1000 | 50.00 | 50.00 | 0.0% |

Pulse counting achieved 100% accuracy for all test cases. The deduction rate was configured at Rs. 0.05 per pulse (Rs. 50 per kWh). Overall balance deduction error was within 2% of the reference meter, attributable to pulse timing jitter acceptable for a prototype-level prepaid metering system.

C. Relay Control Testing

Relay operation was verified under three balance conditions. Results in Table IV confirm immediate relay response.

Table IV: Relay Operation Test Results

| Balance Status | Relay State | Load Status | Response Time |
|---|---|---|---|
| Balance > Threshold | On (Energized) | Connected | < 50 ms |
| Low Balance | On (Energized) | Connected | N/A |
| Balance = Zero | Off (Released) | Disconnected | < 100 ms |
| After Recharge (Zero) | On (Energized) | Reconnected | < 200 ms |

The relay de-energized within 100ms of balance reaching zero, disconnecting the load within one interrupt service cycle. Power restoration after RFID recharge was immediate. The test was repeated 10 times with consistent results, confirming relay reliability.

D. GSM Notification Performance

SMS delivery was tested across 50 events (25 recharge confirmations, 25 low-balance alerts). Average delivery time was 5.0 seconds. SMS delivery success rate was 98% (49/50 delivered), with one failure attributed to a temporary network outage. The system gracefully handled GSM timeout without disrupting other functions.

E. EEPROM Data Persistence

Power was removed from the system at random intervals during operation to simulate power failures. In all 10 test cases, balance was correctly recovered from EEPROM upon power restoration. Maximum data loss per simulated failure was limited to the deduction corresponding to pulses received since the last EEPROM write an expected and acceptable behavior of interrupt-driven EEPROM writes.

F. Overall System Performance Summary

Table V: System Performance Summary

| Parameter | Result | Status |
|---|---|---|
| RFID Scan to Balance Update | < 200 ms | PASS |
| Relay Cutoff on Zero Balance | < 100 ms | PASS |
| SMS Delivery (Average) | 5.0 seconds | PASS |
| EEPROM Balance Retention | 100% (10/10 cycles) | PASS |
| Invalid Card Rejection | 100% | PASS |
| Low Balance Alert Accuracy | Threshold: Rs. 20 | PASS |
| Balance Deduction Accuracy | Within 2% | PASS |
| SMS Delivery Success Rate | 98% | PASS |

## VII. CONCLUSION

This paper presented the design, implementation, and experimental validation of an RFID Based Prepaid Energy Meter using Raspberry Pi Pico. The system successfully integrates RFID card-based recharge, GSM SMS notifications, EEPROM balance persistence, pulse-based energy tracking, and relay-controlled automatic load disconnection into a compact and cost-effective prototype.

Experimental results confirm 100% RFID detection accuracy, sub-200ms balance update response, relay cutoff within 100ms of balance depletion, 98% GSM SMS delivery success, and 100% EEPROM data retention across simulated power failures. Balance deduction accuracy was within 2% of a calibrated reference energy meter.

Compared to prior RFID-GSM prepaid meters built on 8051 or Arduino UNO platforms [9][10], the Raspberry Pi Pico delivers superior processing performance, dual dedicated UART channels for simultaneous RFID and GSM communication, lower unit cost, and lower power consumption making it better suited for scalable deployment in residential, hostel, rental, commercial, and industrial applications. Future work will extend the system with Wi-Fi/IoT connectivity for cloud-based consumption monitoring, online recharge via UPI payment gateways, tamper detection using current monitoring, three-phase metering support, and LoRaWAN integration for rural area deployment.

## VIII. FUTURE SCOPE

Although the proposed system successfully achieves all stated objectives, the following enhancements are identified for future development:

1. IoT Integration: Addition of Raspberry Pi Pico W's onboard Wi-Fi for real-time cloud-based

consumption monitoring and mobile dashboard access.
2. Online Recharge: Integration with UPI, net banking, and QR-code-based recharge portals, eliminating the need for physical RFID cards.
3. Tamper Detection: Accelerometer-based physical tamper detection and CT-based bypass current monitoring with immediate SMS alerts.
4. Three-Phase Extension: Supporting three-phase metering via three pulse inputs and relay channels, suitable for industrial consumers.
5. Renewable Energy Integration: Bidirectional metering for solar net-metering, crediting balance when excess solar energy is fed to the grid.
6. LoRaWAN Communication: Low-power long-range connectivity for rural area deployment where cellular and internet coverage is unreliable.

## REFERENCES

[1] Brinda Vishal and Jaladi, "Design and Implementation of Smart Energy Meter," International Journal of Electronics and Instrumentation Engineering, vol. 2, no. 3, pp. 982–987, Mar. 2013.

[2] Iqbal, Z., M. Wasim, and T. Mahmood, "Design and Implementation of GSM Based Smart Energy Meter for Home Applications," International Journal of Engineering Research and Applications (IJERA), vol. 4, no. 6, pp. 45–50, 2014.

[3] Choudhury, S., A. Sheikh, and P. Rathod, "Design and Implementation of Smart Digital Energy Meter," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 3, no. 5, pp. 9467–9473, 2014.

[4] Morello, R., C. De Capua, G. Fulco, and S. C. Mukhopadhyay, "A Smart Power Meter to Monitor Energy Flow in Smart Grids: The Role of Advanced Sensing and IoT in the Electric Grid of the Future," IEEE Sensors Journal, vol. 17, no. 23, pp. 7828–7837, Dec. 2017.

[5] Tan, H. G. R., C. H. Lee, and V. H. Mok, "Automatic Power Meter Reading System Using GSM Network," in Proc. 8th International Power Engineering Conference (IPEC), Singapore, 2007, pp. 465–469.

[6] Ashna, K., and S. N. George, "GSM Based Automatic Energy Meter Reading System with Instant Billing," in Proc. IEEE International Conference on Control, Communication and Computing, 2013, pp. 65–71.

[7] Vadda, P., "Design and Deployment of an IoT-Based Smart Energy Management System," International Journal of Advanced Computer Science and Applications, vol. 10, no. 8, pp. 250–256, 2019.

[8] Lizama, C., "Smart Energy Meter for Smart Grid Applications," International Journal of Smart Grid and Clean Energy, vol. 5, no. 4, pp. 312–318, 2016.

[9] Microchip Technology Inc., EEPROM Memory Fundamentals and Applications, Technical Reference Manual, 2021.

[10] Raspberry Pi Foundation, Raspberry Pi Pico and RP2040 Datasheet, Raspberry Pi Documentation, 2024. [Online]. Available: https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf

[11] Priority 1 Design Pvt. Ltd., EM18 RFID Reader Module Datasheet, Technical Documentation, 2023.

[12] SIMCom Wireless Solutions Co. Ltd., SIM800 Series AT Command Manual, Version 1.12, 2023.